

#6
BT

8.15-02

RECEIVED

AUG 13 2002

Technology Center 2100

Attorney Docket No.: 4001-1028
PATENT



IN THE U.S. PATENT AND TRADEMARK OFFICE

Applicant: ENGHAUSER et al. Confirmation No. 4300
Appl. No.: 10/086,865 Group: 2673
Filed: March 4, 2002 Examiner: UNKNOWN
For: METHOD FOR SPECIFYING, EXECUTING AND
ANALYZING METHOD SEQUENCES FOR
RECOGNITION

RECEIVED
JUN 24 2002
Technology Center 2600

LETTER

Assistant Commissioner for Patents
Washington, DC 20231

Date: June 19, 2002

Sir:

Under the provisions of 35 U.S.C. § 119 and 37 C.F.R. § 1.55(a), the applicant(s) hereby claim(s) the right of priority based on the following application(s):

<u>Country</u>	<u>Application No.</u>	<u>Filed</u>
GERMANY	101 10 208.9	March 2, 2001

A certified copy of the above-noted application(s) is(are) attached hereto.

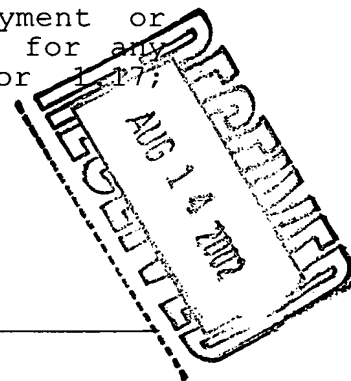
If necessary, the Commissioner is hereby authorized in this, concurrent, and future replies, to charge payment or credit any overpayment to Deposit Account No. 25-0120 for any additional fee required under 37 C.F.R. §§ 1.16 or 1.17; particularly, extension of time fees.

Respectfully submitted,

YOUNG & THOMPSON

By

Robert J. Patch, #17,355



RJP/psf

745 South 23rd Street, Suite 200
Arlington, Virginia 22202
(703) 521-2297

Attachment

BUNDESREPUBLIK DEUTSCHLAND



RECEIVED
AUG 13 2002
Technology Center 2100

Prioritätsbescheinigung über die Einreichung einer Patentanmeldung

RECEIVED
JUN 24 2002
Technology Center 2600

Aktenzeichen: 101 10 208.9
Anmeldetag: 02. März 2001
Anmelder/Inhaber: Siemens Production and Logistics
Systems AG, Nürnberg/DE
Bezeichnung: Verfahren zum Spezifizieren, Ausführen
und Analysieren von Verfahrensabläufen
beim Erkennen
IPC: G 06 K 9/00

**CERTIFIED COPY OF
PRIORITY DOCUMENT**

Die angehefteten Stücke sind eine richtige und genaue Wiedergabe der ursprünglichen Unterlagen dieser Patentanmeldung.

München, den 07. März 2002
Deutsches Patent- und Markenamt

Der Präsident

Im Auftrag

Wallner

Beschreibung

Verfahren zum Spezifizieren, Ausführen und Analysieren von
Verfahrensabläufen beim Erkennen

5

Die Erfindung betrifft ein Verfahren zum Spezifizieren, Ausführen und Analysieren von Verfahrensabläufen beim Erkennen von Sendungsaufschriften und Formulareintragungen.

10

Die Erkennung erfolgt mit Hilfe von OCR-Lesesystemen und Videokodiersystemen, wobei im Falle nicht eindeutiger Leseergebnisse des OCR-Lesesystems die entsprechenden Abbilder der Sendungsaufschriften oder Formulare an Videokodierplätze übertragen werden, wo eine manuelle bzw. teilmanuelle Kodierung mit unterschiedlichen Kodierschritten unter Verwendung von Datenbanken der Aufschriften erfolgt. Die zu lesenden Aufschriften auf den Sendungen sind zum großen Teil Adressen und als Datenbanken werden Adress- oder Nachsendewörterbücher eingesetzt. Mit dem festgelegten Verfahrensablauf als wesentlicher Bestandteil der Ablaufsteuerung wird der Ablauf der Bearbeitung jeder einzelnen Sendung oder jedes Formulars überwacht und gesteuert.

15

20

25

30

35

Der Verfahrensablauf definiert für jede Sendung oder jedes Formular die Abfolge der einzelnen Bearbeitungsschritte und die endgültige Entscheidung, was mit der Sendung oder dem Formular im Gesamtsystem geschehen soll. Bei allen Entscheidungen werden alle für die Entscheidung relevanten Informationen ausgewertet, die diesem Zeitpunkt vorliegen. Die für die Entscheidungen relevanten Informationen sind z. B. Angaben über den Typ der Sortiermaschine, den Typ der einzelnen Kodiergeräte und bisher vorliegende Erkenntnisergebnisse. Wenn alle erforderlichen Schritte ausgeführt sind, wird die Übermittlung des Ergebnisses an die Sortiermaschine veranlasst, damit die Sendung oder das Formular entsprechend diesem Ergebnis verteilt werden kann.

Bisher wurde zur Beschreibung und Realisierung gattungsgemäßer Verfahrensabläufe ein regelbasierter Ansatz eingesetzt. In Abhängigkeit vom Anwendungsfall können daher Hunderte von Regeln zur Beschreibung aller möglichen Abläufe notwendig sein. Diese umfangreichen Regelwerke waren jedoch nur sehr schwer wartbar und selbst kleine Änderungen hatten unvorhersehbare Auswirkungen oder führten zu einer erheblichen Steigerung der Anzahl der Regeln. Die Spezifizierung und die Analyse der Regeln auf Konsistenz und Vollständigkeit erforderte einen sehr hohen Aufwand.

Der Erfindung liegt daher die Aufgabe zugrunde, ein Verfahren zu schaffen, mit dem der Aufwand beim Spezifizieren und Analysieren gattungsgemäßer Verfahrensabläufe stark reduziert wird. Erfindungsgemäß wird die Aufgabe durch die Merkmale des Anspruchs 1 gelöst.

Durch die erfindungsgemäße graphische Darstellung als Flussdiagramm der Verfahrensabläufe mit Attributen und Funktionsangaben, die automatische Umwandlung in ein ladbares Modul, das entsprechend des jeweiligen Bearbeitungsschrittes aufgerufen wird und die Analyse anhand der übersichtlichen Darstellung wird der benötigte Aufwand zum Spezifizieren und Analysieren der Verfahrensabläufe insbesondere in Testphasen erheblich reduziert.

Vorteilhafte Ausgestaltungen der Erfindung sind in den Unteransprüchen dargestellt.

So ist es vorteilhaft, zusätzlich zu den Attributen auch Wertebereiche und Kommentare einzugeben.

Vorteilhaft ist es auch, in die Attributsdatei zusätzlich zu den aktuellen Werten zugehörige Referenzen zu den Namen, Kommentaren und/oder zu den betreffenden Elementen / Bearbeitungsschritten des Flussdiagramms einzutragen.

In einer weiteren vorteilhaften Weiterbildung wird bei der Analyse in dem dargestellten Flussdiagramm der aktuelle Verfahrensschritt markiert.

- 5 Zur verbesserten Übersichtlichkeit ist es auch vorteilhaft, die Attribute und Kommentare des aktuellen markierten Verfahrensschrittes darzustellen.

- 10 Die vorstehenden vorteilhaften Ausgestaltungen erleichtern das Analysieren der Verfahrensabläufe.

- 15 Vorteilhaft ist es auch, während der Analyse die Parameter der Variablen und/oder Funktionen online zu verändern und in das Flussdiagramm einzutragen.

Nachstehend wird die Erfindung anhand der Zeichnungen in einem Ausführungsbeispiel näher erläutert.

Dabei zeigen

- 20 FIG 1 a und b ein Flussdiagramm (Ausschnitt) als graphische Beschreibung eines Verfahrensablaufes zum Videokodieren einer Sendungsadresse.

- 25 FIG 2 eine Bildschirmdarstellung für eine offline-Analyse zum obigen Flussdiagramm.

- 30 Das hier beschriebene System bietet auf Basis gezeichneter Flussdiagramme eine integrierte graphische Entwicklungsumgebung für die Spezifizierung, Ausführung und Analyse von Verfahrensabläufen zum Erkennen (automatisches Lesen und/oder Videokodieren) von Sendungsaufschriften und Formulareintragen. Diese integrierte graphische Entwicklungsumgebung beinhaltet die graphische Beschreibung der Abläufe, die Erzeugung der Laufzeitumgebung und geeignete Tools für Test und
35 Diagnose der Abläufe.

Im Einzelnen werden folgende Funktionen unterstützt:

1. Erfassung und Änderung der graphischen Beschreibung des
Verfahrensablaufes (hier mit Hilfe des Office Zeichenpro-
gramms VISIO von Microsoft),
2. rechnergestützte automatische Generierung von Programmcode
(hier C++ Kode) aus der graphischen Beschreibung,
3. Tools für Analyse (Diagnose und Debugging) auf der Ebene
der originalen Zeichnung (Debugging und Trace Tools).

Während der gesamten Bearbeitung bleiben die gezeichneten Ab-
laufdiagramme die einzige Programmquelle, d.h. alle generier-
ten Teile stellen nur ein Mittel zum Zweck dar und müssen vom
Anwender nicht gepflegt werden. Die Tools für die Analyse
(Diagnose und Debugging) arbeiten ebenfalls mit einem direk-
ten Zugriff auf die graphische Beschreibung.

Im Folgenden wird als Anwendungsbeispiel der Verfahrensablauf
beim Videokodieren von Sendungsadressen (Kodierstrategie)
dargestellt.

Die Kodierstrategie wird durch eine oder mehrere Zeichnungen
auf Basis von standardisierten Zeichnungselementen in Anleh-
nung an ISO Flussdiagramme beschrieben. Zusätzlich werden in
den Zeichnungen Attribute und Funktionen definiert (z.B. als
spezielle Tabellen). Attribute und Funktionen definieren die
Details des Einflusses einzelner Variablen und deren aktuel-
ler Werte auf den Ablauf der Kodierstrategie.

Attribute werden durch einen Namen und einen Typ definiert.
Es werden die Typen „Text“ und „Zahl“ unterstützt. Zusätzlich
können für jedes Attribut noch ein Wertebereich und ein Kom-
mentar angegeben werden. Der Wertebereich beschreibt die Men-
ge der möglichen Werte eines Attributes und in den Kommenta-

ren sollten die Bedeutungen der Attribute erläutert werden. Dadurch ist es später bei der Analyse möglich, die Kommentare automatisch den aktuellen Werten zuzuordnen, was die Verständlichkeit bei der Analyse enorm erhöht.

5

Attribute werden entweder über das Interface der Kodierstrategie als Input aus dem Rahmenprogramm bereitgestellt oder haben für die Kodierstrategie lokalen Charakter (z.B. für die Speicherung von Zwischenergebnissen).

10

Bei Funktionen werden ebenfalls zwei Arten unterschieden. Entweder sind Funktionen innerhalb der Kodierstrategie definiert (lokale Funktionen) oder sie sind Bestandteil des Rahmenprogramms und werden von der Kodierstrategie mit den aktuellen Parametern aufgerufen.

15

In der FIG 1 wird als Beispiel ein Ausschnitt aus dem Verfahrensablauf (Kodierstrategie) zum Videokodieren von Sendungsadressen dargestellt.

20

Eine Erläuterung hierzu erfolgt in der folgenden Tabelle.

Bezugs- zeichen- Nr.	Beschreibung
1.	Start der Funktion DECISION, das ist die Hauptfunktion dieser Kodierstrategie
2.	Aufruf der Funktion PROC EVENT, diese Funktion ist im Rahmenprogramm definiert (siehe Kommentar 12)
3.	Prüfen, ob Attribut RESULT STATUS gesetzt ist
4.	Sprung zur Fortsetzungsmarke RETURN TO RIC
5.	Aufruf der Funktion CHECK CODING, ermittelt welche weiteren Kodierschritte ausgeführt werden müssen
6.	Prüfen, ob Attribut RESULT STATUS gesetzt ist

7.	Sprung zur Fortsetzungsmarke RETURN TO RIC
8.	Aufruf der Funktion PP DECISION, ermittelt alle Merkmale des Sortierergebnisses
9.	Prüfen, ob Attribut RESULT STATUS = RS_MORE CODING d.h. prüft ob weitere Kodierschritte notwendig sind
10.	Setzt Attribut RESULT STATUS zurück
11.	Sprung zur Fortsetzungsmarke RETURN TO RIC
12.	Liste der Funktionen des Rahmenprogramms, die aus der Kodierstrategie aufgerufen werden
13.	Fortsetzungsmarke RETURN TO RIC
14.	Aufruf der Funktionen TRACE VAR für die Attribute RESULT STATUS und UPDATE DSU
15.	Aufruf der Funktion POST PROC, bearbeitet Statistiken
16.	Rückkehr zum Rahmenprogramm, anschließend kann die nächste Entscheidung der Kodierstrategie aufgerufen werden

Tab. 1

Die durch das Flussdiagramm vollständig beschriebene Kodier-
strategie wird eingelesen und in eine interne Repräsentation
überführt (durch VISIO mittels Automation). Aus dieser inter-
nen Repräsentation wird Programmcode (C++ Quellcode) für die
Kodierstrategie generiert. Mit Hilfe des C-Compilers wird
dann ein ladbares Modul erzeugt, das vom Rahmen der Kodier-
strategie aufgerufen wird. Die Kodierstrategie wird bei allen
Änderungen der Attribute einer Sendung aufgerufen und der
weitere Ablauf wird neu bestimmt.

Neben dem generierten Code des eigentlichen Verfahrenablaufes
wird zusätzlich Code zur Erzeugung von Traceobjekten gene-
riert, der während der Laufzeit der Kodierstrategie die Auf-
zeichnung von Diagnoseinformationen steuert. Die Traceobjekte
enthalten alle Informationen zu den Attributen (aktuelle Wer-
te, Bezug zu Namen und Kommentaren) und eine eindeutige Refe-
renz zu den ursprünglichen Elementen des Flussdiagramms.

Während die Kodierstrategie ausgeführt wird, werden für jeden Aufruf diese Traceobjekte zusammen mit den Werten der Attribute in einem Tracepuffer pro Sendung abgelegt. Wenn die Tracefunktion für die Kodierstrategie aktiviert ist, werden die Tracepuffer für eine spätere Analyse in einer Attributsdatei gespeichert, nachdem eine Sendung durch das Kodiersystem vollständig bearbeitet wurde. Ist die Tracefunktion nicht aktiv, dann werden die Tracepuffer nicht gespeichert.

- 10 Zu dem hier betrachteten Beispiel (FIG 1) sind nachfolgend Ausschnitte des generierten C++ Codes dargestellt.

Bezugs-
zeichen-

15 Nr.

```

/*
 * generated code using template template.cpp,
 * do not edit
 */

#ifndef CS_TEMPLATE_INCLUDED
#define CS_TEMPLATE_INCLUDED
...
#include "tracebuf.h"

1 void Decision(void)
  {
    TraceEntry(37,7,"Decision");
    /* The main entry in this CodingStrategy */
    TracePt(37,29);
    ProcEvent();
    2 if (ResultStatus.is_set()) {
    3     TracePt(37,36);
    4     /* Let RIC perform next coding step/ early decision */
    /* Return to RIC */
    TracePt(37,46);
    L1:;
    13     TracePt(37,70);
    14     TraceVar(ResultStatus);
    TraceVar(UpdateDSU);
    TracePt(37,47);
    15     PostProc();
    TracePt(37,49);
    16     /* Let RIC perform next coding step/ decision */
    } else {
    TracePt(37,35);
  }

```

```

L2;;
  TracePt(37,9);
  /* Determine, if additional coding steps are neccessary
before final decision can be made */
5    CheckCoding();
  TracePt(37,16);
6    if (ResultStatus.is_set()) {
      TracePt(37,26);
      ...
7      goto L1; /* 37,46 */
    } else {
      TracePt(37,18);
      ...
8      PPDecision();
      TracePt(37,25);
9      if (ResultStatus == RS_MoreCoding) {
        TracePt(37,27);
10       ResultStatus.clear();
        ...
        goto L2; /* 37,35 */
      } else {
11       TracePt(37,24);
        ...
        goto L1; /* 37,46 */
      }
    }
  }
  TraceExit("Decision"); /* 37,7 */
}

```

Die in der Attributsdatei aufgezeichneten Tracepufferdaten können für die detaillierte Analyse des durch die Kodierstrategie gesteuerten Ablaufes der Bearbeitung jeder Sendung genutzt werden.

Über ein spezielles User Interface werden dem Anwender die Informationen aus dem aufgezeichneten Tracepuffer zusammen mit dem zugehörigen originalen Flussdiagramm der Kodierstrategie dargestellt.

Bei der Offline Analyse kann der Ablauf aller Bearbeitungsschritte einer Sendung nachvollzogen werden. Dabei werden dem Anwender die ausgeführten Schritte in dem originalen Flussdiagramm der Kodierstrategie mit allen für die Analyse relevanten Informationen dargestellt. Für die Analyse der Abläufe

einer Kodierstrategie werden dem Anwender gemäß FIG 2 folgende Informationen in einzelnen Fenstern dargestellt:

- Zeichnung des Flussdiagramms, die aktuelle Position innerhalb der originalen Zeichnung der Kodierstrategie wird markiert, 20,
- das Bild der Sendung, 21,
- Informationen zu den Attributen (bezogen auf den Bearbeitungsschritt werden die jeweils aktuellen Werte der Attribute dargestellt), 22,
- Aufrufliste bei Funktionsaufrufen, 23,
- Kapitelindex, Nummer des aktuellen Kapitels der Zeichnung (Kapitel und Unterkapitel der aktuellen Position) und Liste aller Kapitel, 25,
- optional zusätzliche Logfile Ausgaben, 24.

Der Anwender hat nun die Möglichkeit, den im Tracepuffer gespeicherten Ablauf der Bearbeitung der jeweiligen Sendung nachzuvollziehen. Dabei hat er wie mit einem Source-Level Debugger die Möglichkeit, durch den Tracepuffer zu navigieren.

Es werden die üblichen Kommandos eines Source-Level Debugger unterstützt:

- Benutzung von Break-Punkten,
- Step-In,
- Step-Over,
- Step-Out usw.

In der graphischen Darstellung der Kodierstrategie (VISIO-Flussdiagramm) wird der jeweils aktuelle Schritt markiert.

Der Anwender sieht, wie die Markierung mit der Ausführung der einzelnen Schritte durch die Zeichnung geführt wird. Bei jedem Schritt werden für alle Attribute Werte sowie Kommentare zu den aktuellen Werten dargestellt. Damit stehen dem Anwender für jede aufgezeichnete Sendung alle zur Analyse der Abläufe der Kodierstrategie notwendigen Informationen zur Verfügung, d.h. alle Abläufe können nachträglich im Detail analysiert werden.

Dieses Verfahren bietet die Möglichkeit einer detaillierten Analyse bezogen auf jede einzelne Sendung, die in dem System bearbeitet wurde. Auch nachdem eine sehr große Anzahl von Sendungen bearbeitet wurde, kann die Analyse noch Stunden nach der eigentlichen Bearbeitung der Sendungen vorgenommen werden. Mit diesem Diagnosewerkzeug können selbst hoch komplexe Kodierstrategien analysiert werden und im Falle von fehlerhaften Entscheidungen auf diesem Weg die speziellen Fehlerursachen ermittelt werden.

Das beschriebene Verfahren ist auch für Online Tests geeignet. Bei Online Tests kann eine in der Entwicklung befindliche Kodierstrategie schrittweise ausgeführt und getestet werden. Dabei kann der Anwender über das User Interface der Entwicklungsumgebung mit einem speziellen Kommando in der bezeichneten Kodierstrategie einen Break-Punkt definieren. Bei Erreichen des Break-Punktes wird das Bild der gerade bearbeiteten Sendung im User Interface aufgeschaltet und der aktuelle Tracepuffer wird wie zuvor beschrieben benutzt, um den Ablauf bis zur aktuellen Situation darzustellen und detailliert zu analysieren. Anders als bei der Offline Analyse können hierbei zusätzlich die Attribute auch verändert werden und damit der weitere Ablauf beeinflusst werden. Nach der Analyse kann die Bearbeitung fortgesetzt werden (eventuell bis zum nächsten Break-Punkt).

Patentansprüche

- 5 1. Verfahren zum Spezifizieren, Ausführen und Analysieren von
Verfahrensabläufen beim Erkennen von Sendungsaufschriften
und Formulareintragungen, d a d u r c h g e k e n n -
z e i c h n e t , d a s s in der Spezifizierungsphase die
10 Verfahrensabläufe als Flussdiagramme mit Attributen und
Funktionsangaben, die den Einfluss einzelner Variabler auf
den Verfahrensablauf definieren, in ein Computersystem in
graphischer Form eingegeben werden, wobei aus den Fluss-
diagrammen automatisch Codes als interne Repräsentation
generiert werden, die mittels eines Compilers in ein lad-
bares, ausführbares Modul überführt wird, das in der Abar-
15arbeitungsphase zum Erkennen der jeweiligen Sendungsauf-
schrift oder Formulareintragung für jeden Bearbeitungss-
schritt aufgerufen wird, dass in der Abararbeitungsphase zum
Erkennen für jeden Bearbeitungsschritt Informationen zur
Beschreibung des jeweils aktuellen Ablaufs und die aktuel-
20 len Werte der Attribute in eine Attributsdatei geschrieben
werden und dass in der späteren Analysephase zur Rekon-
struktion der einzelnen Bearbeitungsschritte während der
Abararbeitungsphase der Verlauf der Bearbeitung jeder Sen-
dung oder jedes Formulars in den Flussdiagrammen mit den
25 Attributen zusammen mit den dazugehörigen Abbildern der
Sendungsaufschriften oder Formulareintragungen dargestellt
werden, wobei die Attribute die aktuellen Werte aus der
Attributsdatei für die jeweilige Sendungsaufschrift oder
Formulareintragung zeigen.
30
2. Verfahren nach Anspruch 1, d a d u r c h g e k e n n -
z e i c h n e t , d a s s zu den Attributen zusätzlich
noch Wertebereiche und Kommentare eingegeben werden.
- 35 3. Verfahren nach Anspruch 1, d a d u r c h g e k e n n -
z e i c h n e t , d a s s in die Attributsdatei zusätz-
lich zu den aktuellen Werten der Attribute Referenzen zu

den dazugehörenden Namen und Kommentaren eingetragen werden.

- 5 4. Verfahren nach Anspruch 1 oder 3, d a d u r c h g e -
z e i c h n e t , d a s s in die Attributsdatei zusätz-
lich zu den aktuellen Werten der Attribute jeweils eine
Referenz zu den betreffenden Elementen des Flussdiagramms
eingetragen wird.
- 10 5. Verfahren nach Anspruch 1, d a d u r c h g e k e n n -
z e i c h n e t , d a s s bei der Analyse in dem darge-
stellten Flussdiagramm der aktuelle Verfahrensschritt mar-
kiert wird.
- 15 6. Verfahren nach Anspruch 5, d a d u r c h g e k e n n -
z e i c h n e t , d a s s bei der Analyse die Attribute
und Kommentare des aktuellen, markierten Verfahrensschrit-
tes dargestellt werden.
- 20 7. Verfahren nach Anspruch 1, d a d u r c h g e k e n n -
z e i c h n e t , d a s s während der online Analyse zum
Testen von Abläufen die Werte der Variablen und/oder Para-
meter der Funktionen online verändert werden und dadurch
der Ablauf beeinflusst wird.

Zusammenfassung

Verfahren zum Spezifizieren, Ausführen und Analysieren von
Verfahrensabläufen beim Erkennen

5

Die Erfindung betrifft ein Verfahren zum Spezifizieren, Aus-
führen und Analysieren von Verfahrensabläufen beim Erkennen
von Sendungsaufschriften und Formulareintragungen. In der
Spezifizierungsphase werden die Verfahrensabläufe als Fluss-
10 diagramme mit Attributen und Funktionsangaben, die den Ein-
fluss einzelner Variabler auf den Verfahrensablauf definie-
ren, in ein Computersystem in graphischer Form eingegeben.
Aus den Flussdiagrammen werden dann automatisch Codes als in-
terne Repräsentation generiert, die mittels eines Compilers
15 in ein ladbares, ausführbares Modul überführt wird, das in
der Abarbeitungsphase zum Erkennen der jeweiligen Sendungs-
aufschrift oder Formulareintragung für jeden Bearbeitungss-
schritt aufgerufen wird. In der Abarbeitungsphase zum Erken-
nen werden für jeden Bearbeitungsschritt Informationen zur
20 Beschreibung des jeweils aktuellen Ablaufs und die aktuellen
Werte der Attribute in eine Attributsdatei geschrieben. In
der Analysephase wird der Verlauf der Bearbeitung jeder Sen-
dung oder jedes Formulars in den Flussdiagrammen mit den At-
tributen zusammen mit den dazugehörigen Abbildern der Sen-
25 dungsaufschriften oder Formulareintragungen dargestellt.

1/3

3 DECISION CODE

3.1 MAIN FLOW

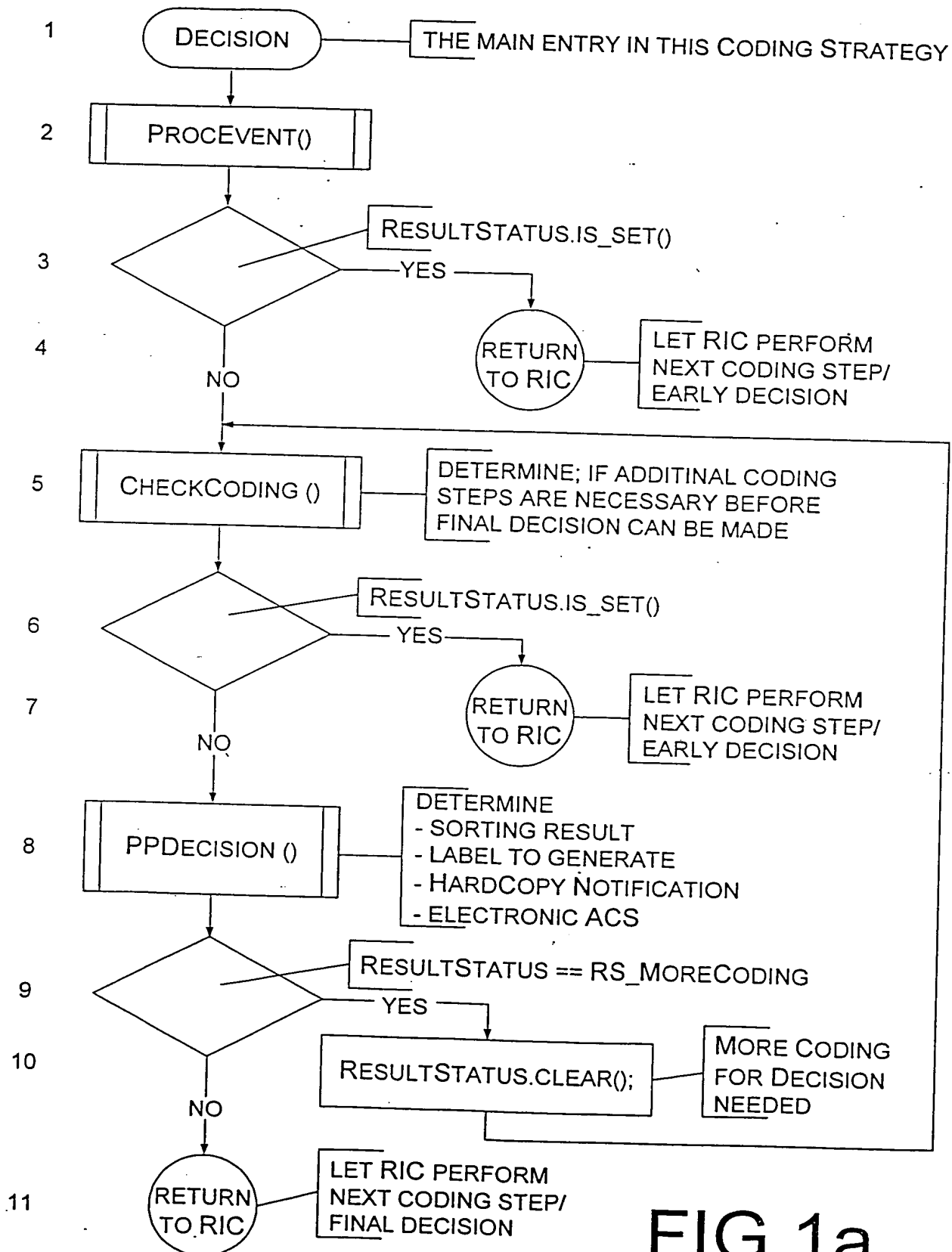


FIG 1a

2/3

12

```
#INCLUDE "INTERFACE.CPP"

VOID PROCEVENT();           //CODING
VOID CHECKCODING();         //CODING
VOID UAA_DUETOMOVE();       //UAA_MOVE
VOID DECIDENIXIE();         //NIXIE
VOID POSTPROC();            //POST_PROC
```

13

RETURN
TO RIC

14

TRACEVAR (RESULTSTATUS);
TRACEVAR (UPDATEDSU);

15

POSTPROC()

DO SOME STATISTIC

16

LET RIC PERFORM NEXT
CODING STEP/DECISION

FIG 1b

BRIEFBILD 21

ATTRIBUTE
(NAMEN UND
AKTUELLE
WERTE)

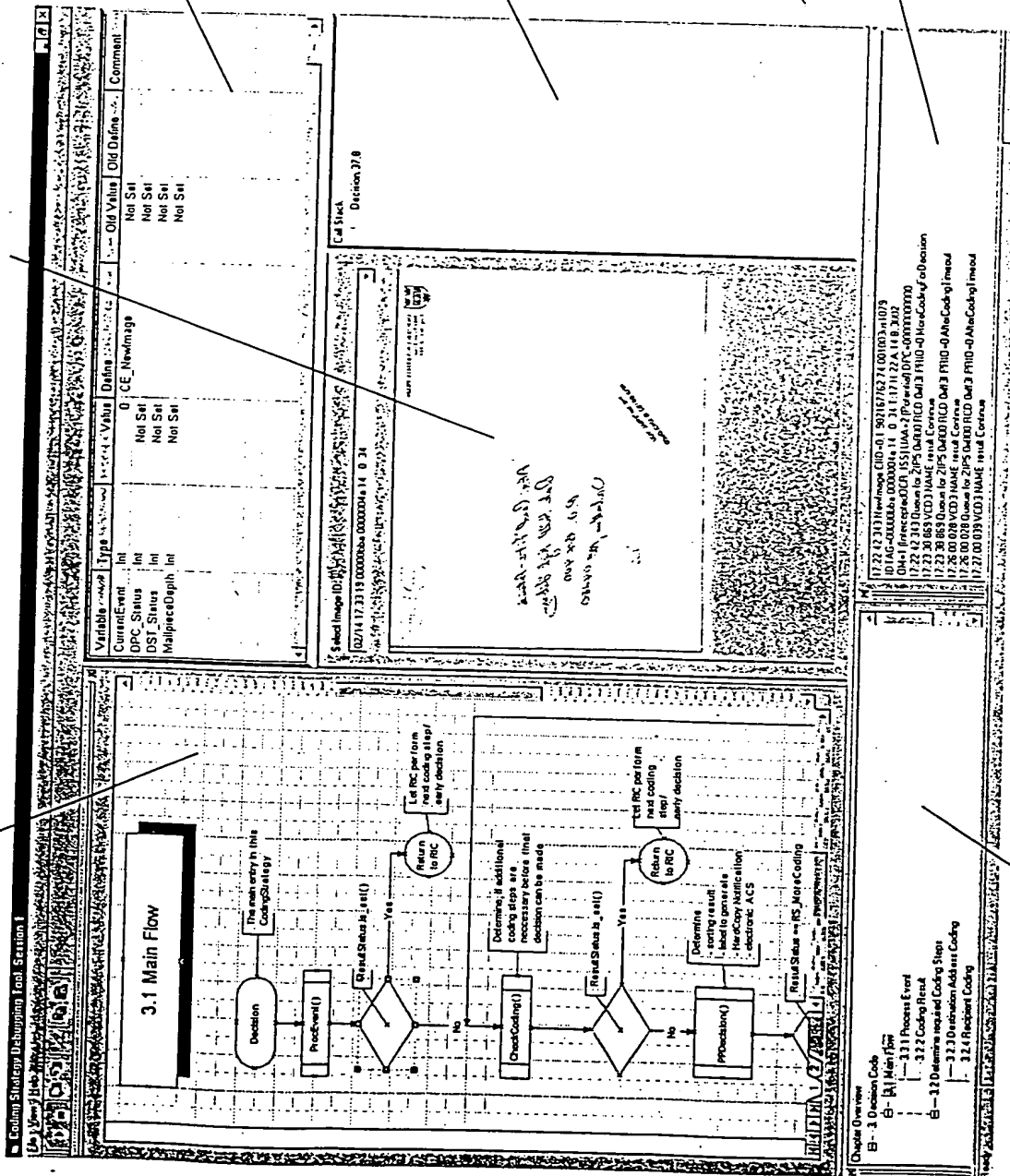
22

AUFLISTE DER FUNKTIONEN

23

AUSGABEN IM LOGFILE

24



KAPITELINDEX / 25

- AKTUELLES KAPITEL
- LISTE ALLER KAPITEL

FIG 2